CLAIMS

1. A computer system for providing a gateway between a transaction manager for managing database transactions from a set of applications, and a server, the computer system comprising

a listener process for receiving inbound connection requests from the transaction manager,

a set of gateway agents for establishing connections to the transaction manager for receiving transaction requests and for establishing connections to the server,

a wait queue,

a logical agent scheduler for managing sets of logical agents, a logical agent comprising data representing a connected application from the transaction manager, each logical agent having an associated inbound connection identifier,

whereby the logical agent scheduler passes a selected logical agent and an associated inbound connection identifier to an available gateway agent and where the gateway has no available gateway agent, providing the logical agent and the associated inbound connection identifier to the wait queue,

a selected gateway agent removes a logical agent and an associated inbound connection identifier from the wait queue when the selected gateway agent is available and the wait queue is non-empty, and

for a logical agent passed by the logical agent scheduler or removed from the wait queue, the gateway agent establishes a connection to the transaction manager as defined by the associated inbound connection identifier and establishes a connection to the server to implement the logical agent.

2. The computer system of claim 1 in which the gateway implements tightly coupled XA transactions from the set of applications by dedicating a single gateway agent to any given tightly coupled XA transaction, the listener process, the logical agent scheduler and the gateway agents passing logical agents to gateway agents such that any logical agent representing a transaction in that given tightly coupled XA transaction will be executed by a given gateway agent dedicated to that given tightly coupled XA transaction.

3. The computer system of claim 2, further comprising a free agent list indicating which gateway agents are available for connection to the transaction manager and which are not dedicated to any given tightly coupled XA transaction.

4. The computer system of claim 2 implemented in a UNIX-based environment in which the connections to the transaction manager are TCP/IP socket pairs and in which the passing of logical agents and associated inbound connection endpoint identifiers is implemented by the use of domain socket pairs in the gateway.

5. The computer system of claim 4 in which the wait queue is implemented as a domain socket pair in the gateway.

6. The computer system of claim 5 in which domain sockets are assigned such that the logical agent scheduler has a domain socket pair dedicated for receiving logical agent and associated inbound connection pairs from the gateway agents and the listener process, and the domain socket pairs for other communication in the gateway are obtained from a pool of domain sockets,

whereby in the case that no domain socket pair is available in the domain socket pool for transfer of a logical agent and associated inbound connection pair the logical agent scheduler will place the logical agent and associated inbound connection pair in the domain socket pair which implements the wait queue.

7. A gateway for demultiplexing connections from a first system to a second system, the gateway comprising internal processes which are selectively connected to implement the demultiplexing function of the gateway, the gateway comprising a wait queue, the wait queue providing a buffering function for both the connections between the first system and the second system and for the connections between the internal processes.

8. The gateway of claim 7 in which the connections between the first system and the second system are TCP/IP socket pairs and the connections between the internal processes are domain socket pairs.

9. The gateway of claim 8 in which the wait queue is implemented by a domain socket pair.

10. A computer system for demultiplexing a set of TCP/IP inbound connections to a set of outbound connections, the computer system comprising a plurality of scheduler processes for providing TCP/IP inbound connections to agent processes for establishing corresponding outbound connections, the scheduler processes and the agent processes communicating by domain socket pairs in the computer system, each scheduler process having a dedicated domain socket pair for receiving a TCP/IP inbound connection endpoint, the domain socket pairs for communication to the agent processes being available from a pool of domain sockets.

11. The computer system of claim 10 further comprising a wait queue implemented as a domain socket pair, the wait queue receiving a TCP/IP inbound connection endpoint where no agent process is available for implementing the TCP/IP inbound connection and from which non-empty wait queue an available agent process will remove a TCP/IP inbound connection endpoint to establish a TCP/IP inbound connection and an outbound connection.

12. A computer program product for use with a computer comprising a central processing unit and random access memory, said computer program product comprising a computer usable medium having computer readable code means embodied in said medium providing a gateway between a transaction manager for managing database transactions from a set of applications, and a server, said computer program product comprising:

computer readable program code means for implementing a listener process for receiving inbound connection requests from the transaction manager,

computer readable program code means for implementing a set of gateway agents for establishing connections to the transaction manager for receiving transaction requests and for establishing connections to the server,

computer readable program code means for implementing a wait queue,

computer readable program code means for implementing a logical agent scheduler for managing sets of logical agents, a logical agent comprising data representing a connected application from the transaction manager, each logical agent having an associated inbound connection identifier,

whereby the logical agent scheduler passes a selected logical agent and an associated inbound connection identifier to an available gateway agent and where the gateway has no available gateway agent, providing the logical agent and the associated inbound connection identifier to the wait queue,

a selected gateway agent removes a logical agent and an associated inbound connection identifier from the wait queue when the selected gateway agent is available and the wait queue is non-empty, and

for a logical agent passed by the logical agent scheduler or removed from the wait queue, the gateway agent establishes a connection to the transaction manager as defined by the associated inbound connection identifier and establishes a connection to the server to implement the logical agent.

13. The computer program product of claim 12 in which computer readable program code means implements tightly coupled XA transactions from the set of applications by dedicating a single gateway agent to any given tightly coupled XA transaction, the listener process, the logical agent scheduler and the gateway agents passing logical agents to gateway agents such that any logical agent representing a transaction in that given tightly coupled XA transaction will be executed by a given gateway agent dedicated to that given tightly coupled XA transaction.

14. The computer program product of claim 13 further comprising computer readable program code means for implementing a free agent list indicating which gateway agents are available for connection to the transaction manager and which are not dedicated to any given tightly coupled XA transaction.

15. The computer program product of claim 12 for implementation in a UNIX-based environment in which the connections to the transaction manager are TCP/IP socket pairs and in which computer program product

the passing of logical agents and associated inbound connection endpoint identifiers is implemented by the use of domain socket pairs in the gateway,

the wait queue is implemented as a domain socket pair in the gateway,

the logical agent scheduler has a domain socket pair dedicated for receiving logical agent and associated inbound connection pairs from the gateway agents and the listener process, and

the domain socket pairs for other communication in the gateway are obtained from a pool of domain sockets,

whereby in the case that no domain socket pair is available in the domain socket pool for transfer of a logical agent and associated inbound connection pair the logical agent scheduler will place the logical agent and associated inbound connection pair in the domain socket pair which implements the wait queue.

16. A computer program product for use with a computer comprising a central processing unit and random access memory, said computer program product comprising a computer usable medium having computer readable code means embodied in said medium providing a gateway for demultiplexing connections from a first system to a second system, said computer program product comprising:

computer readable program code means for implementing internal processes in the gateway which are selectively connected to implement the demultiplexing function of the gateway, and

computer readable program code means for implementing a wait queue, the wait queue providing a buffering function for both the connections between the first system and the second system and for the connections between the internal processes.

17. The computer program product of claim 16 in which the connections between the first system and the second system are TCP/IP socket pairs and the computer readable program code means for implementing the connections between the internal processes implements such connections by using domain socket pairs.

18. The computer program product of claim 17 in which computer readable program code means for implementing a wait queue uses a domain socket pair to implement the wait queue.

19. A computer program product for use with a computer comprising a central processing unit and random access memory, said computer program product comprising a computer usable medium having computer readable code means embodied in said medium providing a computer system for demultiplexing a set of TCP/IP inbound connections to a set of outbound connections, said computer program product comprising:

computer readable program code means for implementing a plurality of scheduler processes for providing TCP/IP inbound connections to agent processes for establishing corresponding Outbound connections, the scheduler processes and the agent processes communicating by domain socket pairs in the computer system, each scheduler process having a dedicated domain socket pair for receiving a TCP/IP inbound connection endpoint, the domain socket pairs for communication to the agent processes being available from a pool of domain sockets.

20. The computer program product of claim 19 further comprising computer readable program code means for implementing a wait queue implemented as a domain socket pair, the wait queue receiving a TCP/IP inbound connection endpoint where no agent process is available for implementing the TCP/IP inbound connection and from which non-empty wait queue an available agent process will remove a TCP/IP inbound connection endpoint to establish a TCP/IP inbound connection and an outbound connection.